BRC Talk 2 – 15 Minutes

How to Rapidly Propagate Accurate Annotations to Closely-Related Genomes

By

Ross Overbeek

In this short talk I will propose that we establish a framework in which users can present us with a genome (either complete or in the form of hundreds of contigs), and we rapidly propagate annotations to it. The goal would be to return a largely-annotated genome within 1-2 days (I am discussing the genomes of bacterial pathogens), with sections requiring more study clearly delimited.

I consider the difficult part of this task to be the development of the accurate annotations to be propagated. I will not discuss the process of developing carefully annotated families of orthologs, since I presume we all are fairly deeply into that task already. I doubt that it makes sense to talk about rapid propagation before we develop initial versions of these families. To make the discussion concrete, let us assume that the following tables (relations) exist for any set of families we wish to propagate:

Family[family_id,genome,gene_id]
FamilyFunction[family_id,function**]**
DnaSeq[genome,gene_id,DNA-sequence]
ProteinSeq**[**genome**,**gene_id,protein-sequence**]**

The last table would contain entries only for genes that encode protein sequences (not for RNA-encoding genes, for example). In addition, I suggest we add

LargeRepetiveElement[LREid,genome,gene_id]
LRE_function[LREid,function]

These last two tables would include data to support recognition of prophages, insertion sequences, and transposition events. The use of the term “large repetitive element” may not be perfect. In this context, all it is intended to mean is “genes that may occur an arbitrary number of times (0 or more)”. Together, these six tables capture enough to allow rapid annotation. I will refer to the six tables together as a **Propagatable Set of Families (PSF).**

I believe that individually a number of BRCs are implementing the tools needed to accurately annotated a newly-sequenced genome, assuming that it is phylogenetically close to a set of genomes encoded in a PSF. I would suggest that it makes sense to jointly develop the tools, or at least to compare the results in order to develop overall improvement. To facilitate these interactions, we should first agree on the function of each tool, most notably the input and output specifications. This should, I believe, be trivial (although I have certainly been overly optimistic before).

I would suggest roughly the following as a start (assuming a Unix command line environment):

propagate_annotations PSF contigs GFF3-annotations UpdatedPSF [parameters]

where input would be

PSF would a directory containing six files, each containing a 2-column tab-separated table

contigs is a fasta file of contigs

and output would be

GFF3-annotations for the new genome

UpdatedPSF an updated version of the PSF

The tool would a GFF3-encoded version of the annotated genome. The optional parameters would usually include things like a prefix to be used in assigning unique gene IDs.

The tool would most likely proceed in something like the following steps:

1. The families would be propagated, where possible (this should get RNAs, as well as a subset of the CDSs).
2. The LREs would be propagated.
3. The genes identified by the families would be used as an training set for glimmer, and glimmer would be used to produce an estimate of the genes present (probably a superset).
4. Glimmer calls corresponding to genes identified in either step 1 or step 2 would be discarded.
5. Optionally, one might blast the relatively few remaining calls against a non-redundant protein database.
6. A post-processing step to remove significant overlaps or occasionally resolve choices (based on similarities).
7. The output would be generated.

The output from such a tool could be used by BRCs or simply returned to the user. It should be clear in the GFF3-annotations which genes were called based on the PSF and which were called based on glimmer extensions.
There is certainly more that could be said about tools to support development of the actual PSFs. Again, I would suggest that it might make sense to cooperate in the development of such tools.

We have developed initial versions of tools to generate the PSF (but not the LREs yet) and a tool to propagate the PSF to a new genome. I am sure that it can be improved, and unless someone gives us a better tool, it will be.

Let me now finish off with a short discussion of some more interesting possibilities – the ability to produce reasonably accurate metabolic reconstructions and inventories of virulence-related genes in the same 2-day time period. This can be done once the appropriate subsystems have been defined and populated with existing, well-annotated genomes. This situation should exist within the next 12-18 months. So, how might we add the projection of subsystems to a rapid annotation effort?

First, we would need to expand the contents of the PSFs to include

RoleInSubsystem[Subsystem,FunctionalRole]
FamilyPlaysRole[family_id,FunctionalRole]
LREplaysRole[LREid,FunctionalRole]
ActiveVariant[Subsystem,genome,variant-id]
VirulenceRelated[Subsystem]

The object in propagating subsystems to newly-sequenced genomes is usually to determine which subsystems have operational variants in the new genome. However, with virulence-related subsystems, we wish to simply enumerate which of the genes is present – that is, all combinations of genes should be thought of as an "operational variant".

Once these four tables have been constructed for the set of genomes that make up a PSF, projection of the subsystem information is straightforward. After the basic projection discussed above is completed, it becomes possible to rapidly construct a fairly accurate list of the subsystems that are active (the basis for a detailed metabolic reconstruction) and to compile inventories of virulence-related genes.